(2)

SANDIA REPORT

AD-A240 078

SAND90-2947 • UC-515
Unlimited Release
Printed June 1991

Current Status of the DNA Remote Security Station (RSS)

J. Bryan Pletta

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789

91-07049

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A04
Microfiche copy: A01

SAND90-2947
Printed June 1991

Distribution
Category UC-515

A-1

CURRENT STATUS OF THE DNA REMOTE SECURITY STATION (RSS)

J. Bryan Pletta
Advanced Technology Division
Sandia National Laboratories
Albuquerque, NM 87185

ABSTRACT

The Remote Security Station (RSS) is being developed by Sandia National Laboratories for the Defense Nuclear Agency to investigate issues pertaining to robotics and sensor fusion in physical security systems. This report documents the current status of the RSS program as of October 1990. The RSS system consists of the Man Portable Security Station (MaPSS) and the Telemanaged Mobile Security Station (TMSS) which are integrated by the Operator's Control Unit (OCU) into a flexible exterior perimeter security system. The RSS system uses optical, infrared, microwave and acoustic intrusion detection sensors in conjunction with sensor fusion techniques to increase the probability of detection and decrease the nuisance alarm rate of the system. The program is entering its final year of exploratory development. The major effort during this final year will be to explore a neural network solution to the sensor fusion task.

Table of Contents

FIGURES

Current Status of the
DNA Remote Security Station (RSS)

9/28/90

1.0 Introduction

The Remote Security Station is being developed by Sandia National Laboratories' Advanced Technology Division under sponsorship of the Defense Nuclear Agency (DNA). The program began in mid-FY'87 and is scheduled for completion of exploratory development by the end of FY'91. The current status of the program as well as planned enhancements will be outlined in this report.

1.1 System Need

Present day physical security systems are greatly reliant upon manpower intensive tasks such as perimeter patrols and manual assessment of alarms. This is especially true for situations requiring temporary perimeter security systems. The RSS counters this deficiency in perimeter security by providing greater flexibility for sensor deployment and more efficient use of manpower. Automated sensor platforms will enhance fixed-site security systems by providing remote detection and assessment capabilities.

Many applications are envisioned, both in fixed and temporary sites, in which a robotic security system would prove useful. Portable sensor platforms can be used for temporary replacement of faulty sensors or enhanced protection of vulnerable points in a fixed site Intrusion Detection System (IDS). High-value assets that are present only for a short period of time could be secured in instances where the installation of permanent IDS sensors would be impractical. A mobile system could provide routine patrol capabilities, freeing the human guards from one of the more mundane tasks required in site security. Alarm assessment could be accomplished safely from the security control center and delay or deterrent devices could be activated without placing security personnel in potentially dangerous situations.

1.2 System Philosophy and Overview

The RSS system consists of three major components. These are the Man Portable Security Station (MaPSS), the Telemanaged Mobile Security Station (TMSS) and the Operator's Control Unit (OCU). MaPSS and TMSS are both shown in Figure 1; MaPSS consists of the of the tripod and box in the left side of the photograph, while TMSS is the mobile robot on the right. The OCU is pictured in Figure 2.

Figure 1: Man-Portable Security Station (MaPSS) and
Telemanaged Mobile Security Station (TMSS)

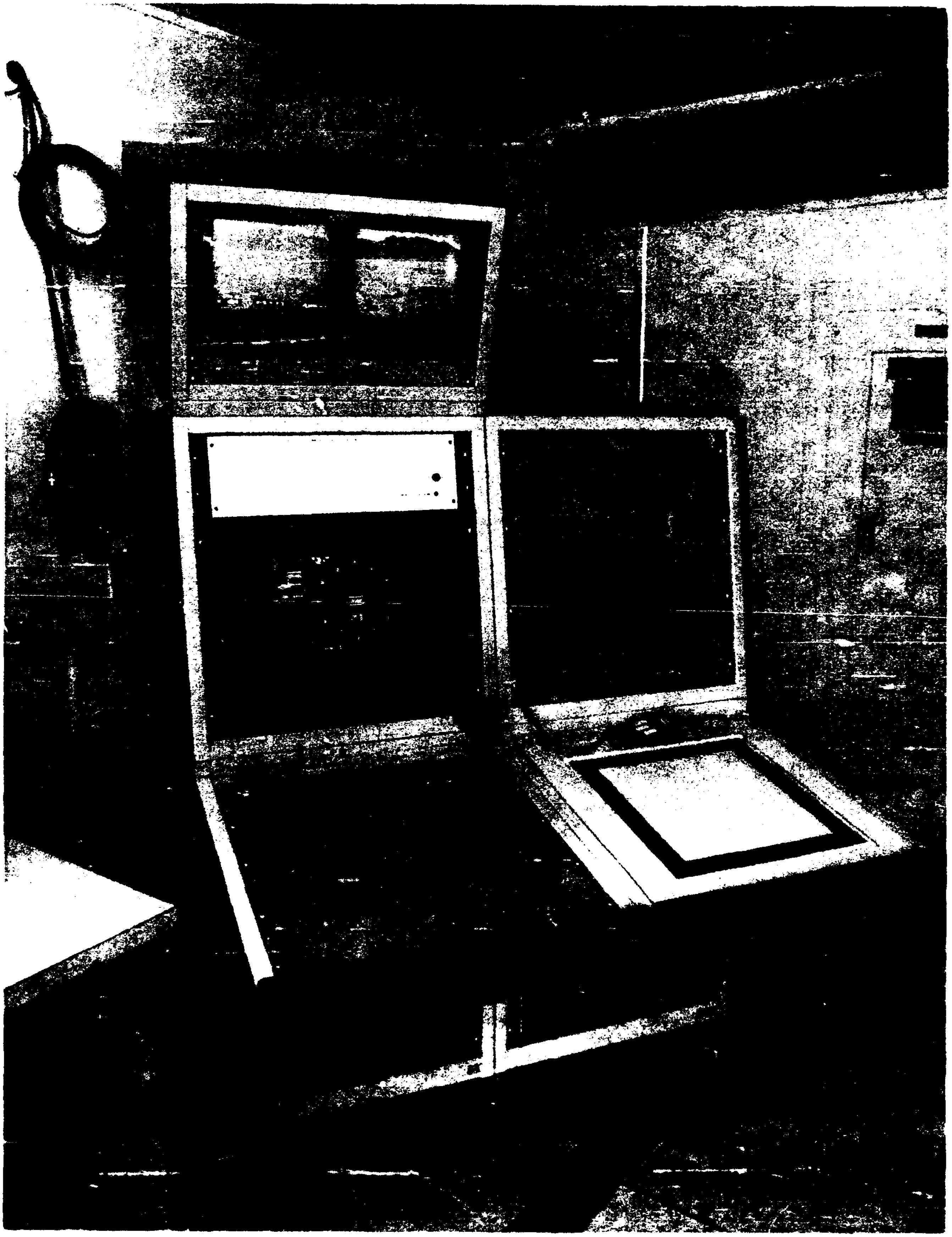

Figure 2: RSS Operator's Control Unit (OCU)

The RSS addresses the need for a flexible perimeter security system by providing both portable and mobile sensor systems that are easily deployed. The MaPSS sensor pod provides a stationary suite of security sensors that is ideal for emplacement around a temporary perimeter, protection of temporary assets, or replacement of faulty sensors. The TMSS sensor pod was developed to add mobility to the MaPSS sensors. This useful addition allows automated patrols and remote assessment of alarms. The OCU provides the integration of security functions and robotic control of the MaPSS and TMSS sensor platforms.

The intent of the program has been to use existing technology when possible. Existing perimeter security sensors have been designed to operate in very structured environments. These sensors are typically mounted on rigid poles between fences so that nuisance alarm sources are reduced to a minimum. The RSS system utilizes additional sensor processing to filter out nuisance sources which are present due to less stringent sensor installation practices.

Raw sensor alarms from TMSS and MaPSS are sent to the RSS OCU where they are combined in a sensor fusion algorithm. The believability of each alarm, based upon the current weather conditions and the sensor's past performance, is reflected in a weight assigned to that alarm. The importance of a site sector is encoded in an alarm threshold. The operator is only notified of alarms if the weighted sum of alarm values exceeds the importance threshold of a sector. The algorithm is designed to increase the probability of detection while reducing the nuisance alarm rate from the sensors and thus reduce the security officer's burden.

2.0 Hardware Description

2.1 MaPSS

The MaPSS sensor pod was primarily developed during FY '88 and has remained essentially unchanged since that time. MaPSS consists of a suite of intrusion detection sensors mounted on a pan/tilt and tripod, and a control box which houses power supplies, communications equipment, and electronics for control. The RSS weather station was originally included in the MaPSS hardware but has recently been moved, so that it feeds directly into the OCU. The MaPSS security sensors include:

- A Cohu black and white CCD video camera with Pelco 16-160mm zoom lens. This lens length corresponds to a horizontal field of view between 43.3 deg and 4.5 deg. This camera is used for assessment as well as detection in conjunction with a Sandia developed Video Motion Detector.

- An Eltec 861 Passive Infrared Motion Sensor.

- An AN/PPS-15 Ground Surveillance Radar.

- An Acoustic Detection, Tracking And Classification System.

- A Dan Gibson Electronic Parabolic Microphone used for audio assessment of alarms.

- A Set Beam near infrared spotlight which is used for covert night time assessment in conjunction with the CCD camera.

The capabilities of these and the TMSS sensors is discussed in section four. Some of these sensors are identical to those used on TMSS while others are different. The sensors themselves were chosen to represent different sensing media and are not necessarily the best example of a particular sensor type.

A custom control system based on the Motorola 6805 microprocessor provides the interface between the OCU and the MaPSS hardware. The 6805 reads alarm relays from the security sensors and communicates this information back to the OCU via a fiber optic serial communication link. It also receives commands from the OCU and implements closed loop control of the pan/tilt and zoom lens functions.

The communication link between MaPSS and the OCU utilizes seven optical fibers. Each of these fibers is used to transmit a single signal between the OCU and MaPSS. Serial data using RS-232 protocol is being transmitted over two of the fibers, one for each direction. In addition to the data links, four audio channels and one video channel are being transmitted from MaPSS to the OCU.

2.2 TMSS

The TMSS portion of the system is basically a duplicate of the MaPSS sensor pod on wheels. A similar set of security sensors and pan/tilt is mounted on an extendable mast that may be raised to a height of 10 feet. The entire sensor package is carried by a mobile robotic vehicle which is based upon a Honda 350 All Terrain Vehicle (ATV). Electric actuators were added to the base vehicle so that its controls could be remotely operated. The TMSS security sensors include:

- A Canon color CCD video camera with Pelco 12.5-75mm zoom lens. This lens length corresponds to a horizontal field of view between 53.9 deg and 9.7 deg. This camera is used for driving and assessment, as well as detection in conjunction with a Sandia developed Video Motion Detector.

- An Eltec 862 Passive Infrared Motion Sensor.

- A Southwest Microwave 375 monostatic microwave sensor.

- A Sandia developed near infrared spotlight which is used for covert night time assessment in conjunction with the CCD camera.

There are two computer systems on TMSS which provide the necessary vehicle controls. A slightly modified version of the MaPSS 6805 controller is used to provide the sensor interfaces and closed loop control of actuators. This processor is referred to as the TMSS-6805. A more powerful processor was recently added to implement high level routines, such as navigation, on-board the robot. This computer is an MS-DOS 16 MHz 80286 industrial computer based on the STD bus and is referred to as the TMSS-AT. The TMSS-AT registers an 18.7 performance index relative to an IBM PC/XT, based on the Norton benchmark speed test.

Communication with TMSS is accomplished via two radio links. A full duplex 1200 baud RF modem operating in the 400MHz frequency range is used for data communications. Video and audio are sent from the vehicle to the OCU over a 900MHz FM video link. The data modems are manufactured by Repco Inc. and the video system is made by Dell Star Inc.

2.3 OCU

The OCU was designed to act primarily as a security officer's interface to the system, with robotic vehicle control acting as a secondary function. Although the OCU was designed for installation in a security control center, it has been deployed in the rear of a step van during field operations.

The OCU is housed in two 19 inch equipment racks with overall dimensions of 74"H x 44"W x 49"D. The total weight is approximately 500lbs. The OCU operates from 120 VAC power and consumes approximately 1400W of power.

The OCU displays consist of two 9 inch black and white video monitors for alarm assessment purposes, a 13 inch color graphics monitor used for map displays, and a 13 inch color computer/video monitor. The computer monitor is equipped with a touch screen and is used as the primary operator's interface to the system. It can also be switched to a video mode to be used as a driving monitor. A digitizing tablet is included for drawing site maps on the graphics display.

The host computer is a 20 MHz 80386 based IBM-PC compatible machine which contains 2 MB of memory, an 80387 math coprocessor, a 20 MB hard disk, 360 KB and 1.2 MB floppy drives, CGA video

board, nine RS-232 serial ports, sixteen A/D channels, and two eight bit I/O ports. At this time, only six of the serial ports and four A/D channels are being used. This computer is responsible for handling the graphic displays, user interface, communications, sensor fusion, robot control, and data logging functions.

A Video Motion Detection (VMD) system which can process video from either TMSS or MaPSS is included in the OCU. The VMD is based on a 68020 microprocessor in a VME bus. There are six additional image processing boards in the VME chassis required to implement the VMD algorithms. This version of the VMD is being used to facilitate algorithm development. A custom hardware version of the system has been built for other applications which considerably reduces the VMD system's size, weight, power, and cost. In addition, the custom hardware version is able to process signals from two video inputs simultaneously.

The computer which processes audio signals for the Sandia developed Acoustic Detection Tracking And Classification System (ADTACS) is also included in the OCU hardware. ADTACS uses signals from three equally spaced microphones on the MaPSS sensor pod to detect acoustic sources, determine their bearing angle, and classify the source. This system is based on a 68020 microprocessor in a VME bus. In addition to the CPU there are additional computer boards in the system for sampling and digitizing the microphone signals.

The weather sensors are attached to the host computer. Wind speed, temperature, light level, and the presence of precipitation are all measured for use in the sensor fusion algorithm.

The primary operator's interface is a graphics display with touchscreen, chosen for its ease of use by untrained operators. The touchscreen is used for alarm indication and assessment, system configuration, and display of system parameters. The vehicle controls are implemented with an aircraft style joystick and an array of switches for special functions.

3.0 Software Description

3.1 MaPSS

The MaPSS software is written in assembly language on the 6805 microprocessor. This software is responsible for communicating with the OCU via RS-232 serial data, closing the control loop on the pan and tilt motors, and reading in sensor data. A block diagram of the MaPSS control program is shown in figure 3.

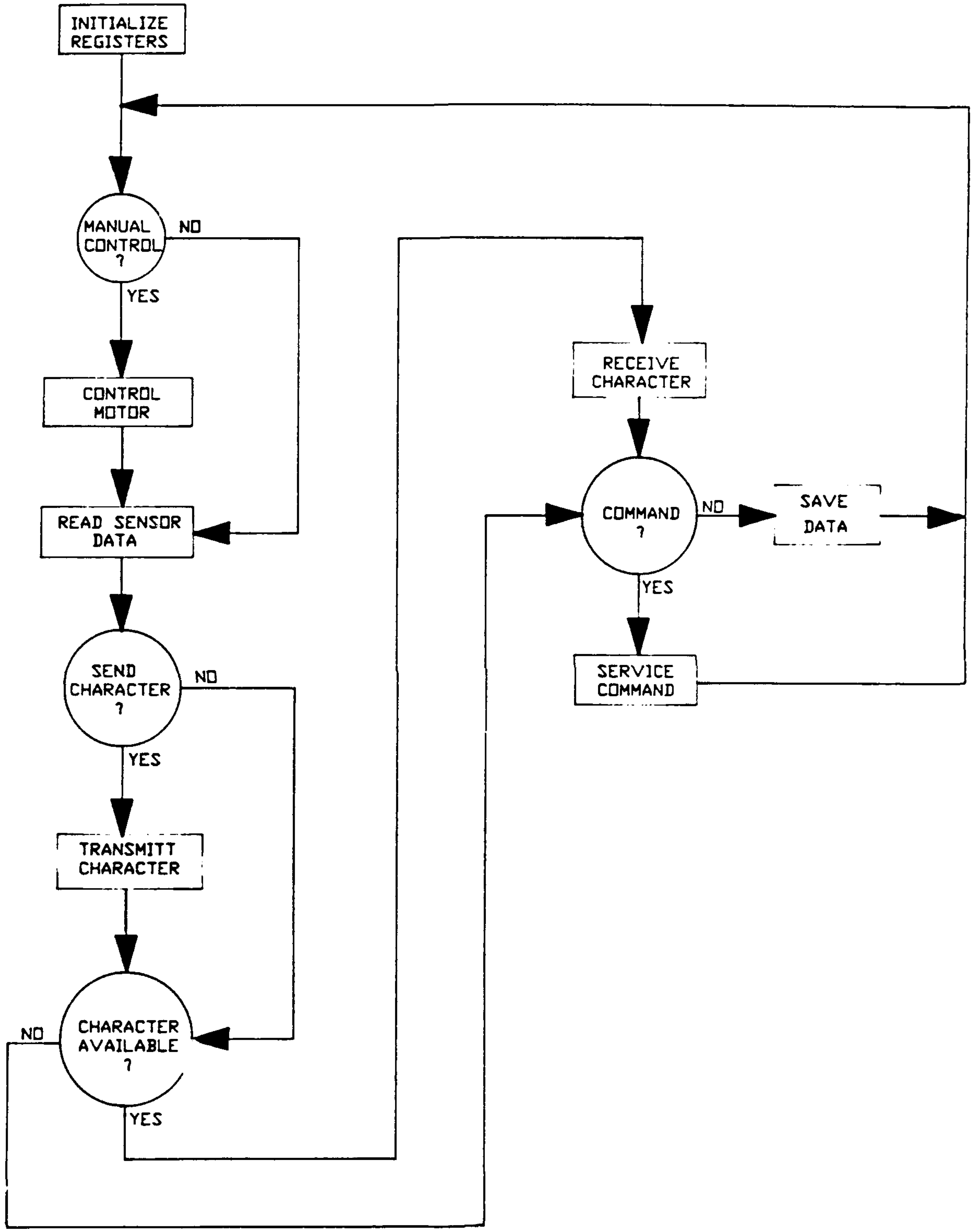

Figure 3. MaPSS Software Flow Diagram

Serial communication with the OCU occurs at 9600 baud in a polled mode. Once per iteration of the main loop, the CPU checks the serial port to see if a character is available for reading and also sends a character out if needed. A complete description of the 6805 communication protocol is given in Appendix A.

Pan and tilt motors are controlled in two different modes. The first is an open loop control mode in which the motor is simply turned on in a given direction upon receiving an operator command. The computer continues to drive the motor in this direction until a stop command is received. The second is a closed loop control mode in which the position of the pan/tilt axes are controlled. The velocity of the pan/tilt motors is slow enough that bang-bang control works effectively. In this control method, the computer compares the commanded position with the actual position of the motor, if the difference between these positions is outside a dead band, the motor is driven in the proper direction to close the gap. Once the position error is less than or equal to the dead band, the motor is stopped.

3.2 TMSS

As noted above, there are two computers on-board TMSS and subsequently two main programs controlling tasks on the vehicle. The vehicle interface and control functions run on the TMSS-6805 microprocessor. This code is written in assembly language and is modelled after the MaPSS software. The high level functions on-board the robot are handled by the TMSS-AT computer. This software is written in C and is running under AMX-86. AMX-86 is a real-time multitasking executive that manages the execution of independent tasks on the TMSS-AT computer. AMX allocates processing time for the individual tasks based on their priority and real-time events occurring within the system.

A block diagram of the TMSS computer system is shown in figure 4. An abbreviated block diagram of the TMSS-6805 software is depicted in the lower right corner of this diagram. There is a main loop executing approximately once every 3 ms that calls a series of subroutines to do closed loop position control of the motors and to read data from the various sensors on board the vehicle. The serial port is checked at the end of this main loop to determine if any new characters are available for processing. If a character is available, it is compared with the list of commands to see if there is a match. If a match occurs, the command subroutine is executed, otherwise the character is assumed to be data needed by a future command and is pushed onto a stack to be retrieved when that command arrives.

A block diagram of the TMSS-AT software is depicted in the upper right corner of figure 4. Some of the tasks have been combined in the diagram for clarity. In reality, the Autonomous

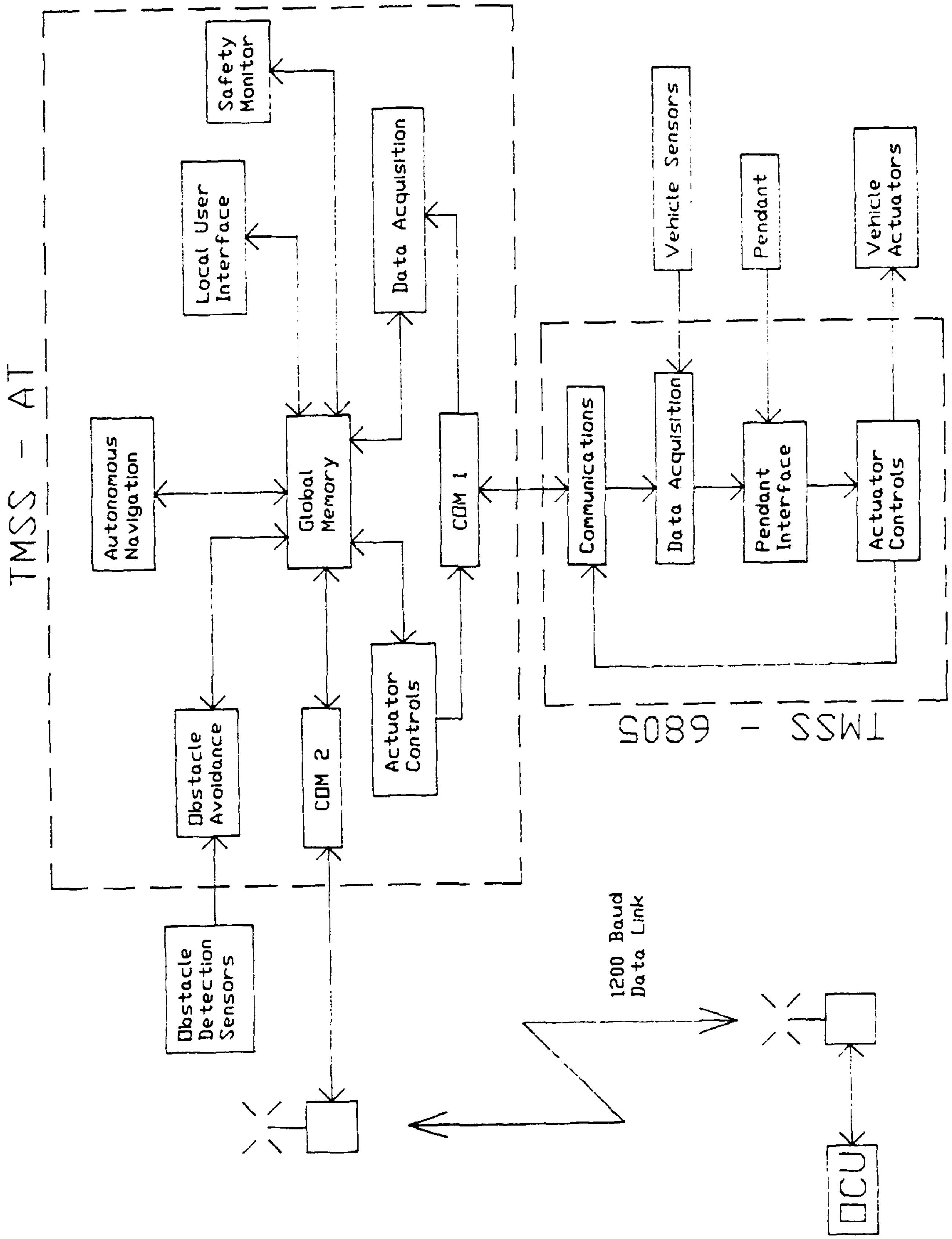

Figure 4. TMSS Computer System

Navigation and Local User Interface tasks are broken down into additional subtasks. This software operates under the AMX multitasking executive and subroutines are not executed sequentially, as would normally occur in standard software. Instead, the code is divided into a set of tasks represented by the boxes in the figure. Each task is given a priority, relative to the others, and an event which triggers that task. The execution of the tasks is managed by AMX, based upon their priorities and the events occurring in the system. This decouples the programmer from timing considerations in the software and makes it easy to add new tasks to the system.

The TMSS-AT computer system is currently in the process of being debugged and implemented, and had not been completed at the time that this report was written. This process is expected to be completed by the end of calender year 1990. The system contains two communication tasks; the COM1 task communicates to the TMSS-6805 computer, while the COM2 task communicates with the OCU. Two subtasks are associated with the COM1 task. The Actuator Controls task is charged with sending commands to the 6805 while the Data Acquisition task receives status information from the 6805. The Actuator Controls task generates actuator commands for the TMSS-6805 computer. These commands can be the result of three separate tasks. In teleoperation mode, the OCU commands are passed through to the TMSS-6805. In autonomous mode, the actuator commands are generated by the Autonomous Navigation task. In either case, the Obstacle Avoidance task has a higher priority than these other tasks and can be used to override actuator commands if an obstacle is seen. The Safety Monitor task looks for error conditions in the system and executes emergency kill procedures or other error handling routines depending on the severity of the error condition. The Local User Interface task is used to facilitate development of the system. It allows a user to view system information on a terminal or input information from a keyboard. Global memory can be accessed by all tasks and is used to pass information between the tasks.

3.3 OCU

The OCU software has been developed with the C programming language because of its flexibility in providing hardware control. Five primary sections make up the OCU control program: communications, data management, sensor fusion, user interface, and control. This program contains a total of approximately 13,000 lines of code including comments. This is estimated to result in approximately 8500 lines of executable code.

The communications software handles serial communications to the graphics display, digitizing tablet, VMD, ADTACS, MaPSS, and TMSS. Only two serial communications interrupts are supported on MS-DOS machines; COM1 has been dedicated to TMSS, while the CC"2

interrupt is shared between the remaining five devices through a special multi-port board. The communications routines handle initialization and shut down of the ports, packaging of data, and handling of the communication interrupts. This section of the software contains 915 lines of code.

The data management software handles logging of data, data conversion, internal tables, linked lists, and other data handling utilities. A total of 2370 lines of code are included in this section of software.

The sensor fusion software processes raw alarms and environmental data to determine system alarms. The raw intrusion alarms are weighted based upon the current environmental conditions. The weighted sum of all raw alarms occurring in a particular sector is then compared against the importance threshold given to that area. A system alarm is generated only if this threshold is exceeded. There are 880 lines of code in this section of the software.

The user interface software is by far the largest section of code in the OCU program, occupying a total of 6430 lines. The user interface code is responsible for generating graphics for both the site map and touchscreen displays, and handling inputs from the digitizing tablet, vehicle controls, and touchscreen.

The final section of code handles system control logic. This involves keeping track of the status of the sensor pods and VMD, making decisions about turning equipment on and off, and generating actuator commands. A total of 2250 lines of code are contained in this section.

4.0 Operational Capabilities

The RSS system currently has the ability to interface two sensor pods to the OCU. At the present time only one copy each of MaPSS and TMSS exist, so the capability for interfacing with one of each was developed. The interface hardcoded into the system requires that TMSS be attached as pod #1 and MaPSS as pod #2. Note that if one of the pods is not attached, the system will not execute the code associated with the missing sensor pod. Control and alarm display for the individual pods is switched via the touchscreen. When both pods are attached, control and display screens for a particular pod are not shown when the other pod is selected. The operator will still be notified of alarms occurring on the other pod with a tone and message prompt on the screen. At this point, the operator may select the other pod as primary to display information and control its actions.

Future enhancements will allow control of up to five sensor pods. These pods will be capable of being either MaPSS or TMSS type in any mix.

4.1 Security Sensors

The RSS intrusion detection sensor suite was chosen to give a representative mix of sensors for the investigation of sensor fusion techniques. Since no requirement for detection ranges was ever given, the chosen sensors have a wide range of maximum detection distances. This ranges from 60m for the Southwest Microwave 375A to 1.5km for the AN/PPS-15 for detection of a walking man. The sensors were chosen to provide detection capabilities over a wide range of the energy spectrum.

4.1.1 Eltec Passive Infrared Motion Sensor (PIMS)

The PIMS detects infrared energy in the 8 - 14 micron range. The sensor has a nominal range of 150m as claimed by the manufacturer, but we have seen detections at longer ranges in favorable conditions. Walkers have been detected at ranges of greater than 200m and heavy construction equipment has been detected at estimated ranges of 500m. The sensor can detect targets with a temperature difference of 1 deg C from the background. The PIMS will detect objects moving across its field of view at speeds of 0.2 m/s to 5.0 m/s. While the sensor is most sensitive to movements across its field of view, it will also detect radial motion.

Two versions of the Eltec PIMS are being used in the RSS system. The original sensors were model 861-01. An upgraded model 862-71 is currently being evaluated. The upgraded sensor is supposed to provide a greater immunity to environmentally induced alarms and provide a curtain coverage as opposed to a line coverage.

Experience with the PIMS has indicated that the sensor has marginal utility for use in an unstructured environment. Most of our experience has been with the line coverage version of the sensor. While this sensor has good range and is relatively insensitive to environmental effects, its major drawback is that it has a limited coverage area. The line coverage version will detect movement in a spot that is only 3.0m wide by 3.6m high at 150m. Although it is ideally suited for determining if an intruder has crossed a well defined perimeter, the sensor lacks the ability to cover a large area outside of the perimeter.

There was hope that the curtain coverage version of the sensor would alleviate this deficiency of the PIMS. This sensor is supposed to detect movement across a curtain spanning from 1.5 degrees above the aim point to 21 degrees below the aim point. Unfortunately, preliminary walktesting has not produced promising results. Instead of providing a curtain of coverage as expected, the sensor seemed to have a pattern much like the narrow beam sensor. The sensor that was tested could possibly be faulty, so

this needs to be investigated further. The walktest was performed at the edge of an arroyo containing quite a bit of vertical relief. The PIMS was designed for installation in fixed site perimeters where the ground is essentially flat and featureless. It seems that the sensitivity is not as great for the lower angles of coverage. While suitable for flat terrain, where low angle targets are close to the detector, the sensor does not seem to provide adequate coverage in the downward direction over varied terrain. This deficiency could possibly be overcome by using an array of sensors spaced vertically to provide a continuous curtain of coverage at the sensor's full range.

4.1.2 Southwest Microwave 375A

The Southwest Microwave model 375A is a monostatic microwave transceiver that detects motion based on the doppler shift principle. The 375A radiates a microwave beam of 10 mW peak power at 10.525 GHz. The unit is capable of detecting an upright man at a distance of 60m. The detection pattern is very nearly a cone shape with a maximum width of 7.3m and height of 5.2m at a distance of 45m. The detection zone then remains essentially constant out to its maximum range of 60m.

The 375A contains a Range Cutoff (RCO) circuit which prevents alarms beyond a user specified distance of 30, 45, or 60m. The RCO circuit turns off the receiver at the appropriate time to ignore reflections from objects past the cutoff distance. While this feature is desirable for fixed site installations, it unnecessarily limits the range for this application.

Discussions with Southwest Microwave have indicated that there is a strong possibility that the range of the sensor could be substantially increased. Simply disabling the RCO feature of the current sensor would increase the range to approximately 85m. Other techniques that reduce the noise figure or increase system sensitivity would increase the range for detection of an upright man to an estimated distance of 200-300m.

4.1.3 AN/PPS-15 Ground Surveillance Radar (GSR)

The GSR is being used in place of the Southwest Microwave sensor on the MaPSS sensor suite. The AN/PPS-15 is a Pulse Frequency Modulated/Continuous wave (PFM/CW) radar with a relatively low power output of 30-94 mW. The unit is designed so that the primary mode of detection is provided by a human operator listening to the doppler audio return from a target. The doppler signal produces characteristic sounds that allow a trained operator to easily identify targets. The unit also has an automatic detection capability that examines returned signal power to determine alarm conditions. This automatic mode is being used with the RSS.

The manufacturer claims maximum detection ranges of 500m for crawling humans, 1500m for upright humans, and 3000m for vehicles. More realistic values for range measurements are approximately 1000m for upright humans and 2000m for vehicles. The limits in detectable radial velocity are listed as 0.5 to 75kmph. Although the return signals are attenuated, higher velocity targets with large returns are detectable.

The automatic detection feature of the GSR is being used to reduce RSS operator workload. Experience with the sensor has shown that this feature is extremely prone to nuisance alarms. It is undesirable to require the operator to constantly monitor the audio feedback of the GSR to manually determine alarm conditions. Therefore, unless automated signal processing of the sensor's audio output is developed, the GSR will not be appropriate for this application. It is hoped that neural network processing of the audio signal will improve the performance of the sensor to an acceptable level.

4.1.4 Video Motion Detection (VMD)

The VMD system is the result of an ongoing development effort at Sandia. The Sandia VMD was chosen over commercial versions so that improvements in the system could be incorporated as they became available. The original VMD was a relatively simple version that detected changes in one dimension only, along an operator defined line of boxes on the video monitor. This system has evolved into a two dimensional system that will detect motion over the entire screen. Other improvements include adaptive thresholding and jitter techniques that reduce environmentally induced nuisance alarms.

The VMD system requires standard RS-170 video as an input, which can be obtained from many thermal imagers and low light cameras in addition to standard video cameras. This feature gives the VMD a great deal of flexibility by allowing it to operate with a choice of two complementary energy sources.

Detection ranges and probabilities are difficult to determine for this sensor. These parameters are extremely dependent on lens length, lighting, and the target's size and contrast with the background scene. With TMSS's zoom lens set to the 75mm length, the VMD can normally detect humans out to a range of at least 200m. This corresponds to a horizontal field of view for the sensor of just under 10 degrees.

Of all the sensors being used in the RSS system, the VMD seems to be the best suited for the application. The VMD combines the advantages of excellent range, good immunity to environmentally induced nuisance alarms, and the ability to work with inputs from

the optical or infrared band. By choosing the appropriate lens length for the camera, a tradeoff between field of view and range can be made to provide the desired characteristics.

4.1.5 Acoustic Detection Tracking And Classification System (ADTACS)

The ADTACS system was developed at Sandia primarily to provide acoustic detection of helicopters. Modifications were made for the RSS system so that propeller aircraft and vehicles could also be identified. Other sources, most notably jets, are ignored. The ADTACS system has a nominal range of about 5km for detection of helicopters, although detections have been made out to approximately 15km in favorable conditions. Its accuracy is approximately +/- 5 degrees in bearing estimation. The range for detection of other aircraft is slightly less than 5km and on the order of 1km for land vehicles with large acoustic signatures.

Experience with ADTACS has shown that it is reliable for helicopter detection, due to the very distinct acoustic signatures that they present. In contrast, the system is susceptible to nuisance alarms classified as vehicles or unknowns because the acoustic signatures are relatively broad band and indistinct. An additional problem seen with the system is that no ranging information is available. Therefore, acoustic sources present outside of the region of interest can cause alarms as well as sources inside the security zone.

4.1.6 Dan Gibson Electronic Parabolic Microphone (EPM)

The EPM was included in the RSS sensor selection to aid in audio assessment of alarms. This microphone replaced the Bionic Ear which was part of the original MaPSS hardware. The EPM has a range of approximately 75m in favorable conditions for listening to conversational speech. Movement and voices can be heard at greater distances, although individual words are generally not comprehended.

None of the directional microphones that were tested seemed to provide a truly useful addition to the RSS capabilities. The amplification of background noises is so significant that the use of a directional microphone seemed to be more of an annoyance than a useful assessment tool. The EPM was chosen as the best directional microphone available after extensive testing of commercial products. The ISOPADS microphone system, developed by the Army's Harry Diamond Laboratory, has a claimed range of 250m for conversational speech and seemed promising as a replacement for the EPM. ISOPADS utilizes fluidic technology and problems were encountered when attempts were made to interface with the system electronically, rendering the system unuseable for the RSS application.

4.1.7 Infrared Spotlight

An infrared spotlight is being used in conjunction with CCD video cameras for covert assessment at night. The CCD cameras are sensitive to near IR energy which is invisible to the human eye. CCD cameras normally have an IR cut filter installed over the image chip to filter out this wavelength of energy. The cut filter has been removed on the RSS cameras so that they are sensitive to the output of the IR light.

Two versions of IR spotlights have been evaluated. The Set Beam is a commercially available near IR spotlight that uses a xenon arc lamp source. The arc provides a high intensity point source of light that is very easily focused. The performance of this light is impressive, scenes up to several hundred meters away can be adequately illuminated with the Set Beam for assessment with the video camera. The drawback of this design is that the arc produces a significant amount of electrical noise. Turning on the Set Beam produced such a large voltage spike on the logic supply of the MaPSS microprocessor that it would occasionally latch up the computer. Shielding of the components as well as filtering and isolation of the power supplies never resolved this problem.

A new light was designed that utilizes a 250W incandescent bulb in an effort to solve the noise problem. Similar to the Set Beam, the lamp generates energy from near IR up through visible light wavelengths. A filter lens which blocks the visible portion of the spectrum is used so that only the IR wavelengths are emitted from the spotlight. There are two characteristics of incandescent bulbs that are undesirable for this application. First, incandescent bulbs produce a line source of light which cannot be focused like the point source produced by an arc lamp, limiting the usable range of the spotlight. Second, incandescent light sources are not as efficient as arc sources and quite a lot of energy is lost as heat. It has been difficult to find an IR filter that can withstand the heat generated by the light without shattering. A new filter was recently purchased from Devoe Aviation that shows promise of working although extensive life cycle tests have not been conducted. The range of this spotlight also has not been determined to date.

4.2 Sensor Fusion

The current sensor fusion algorithm operates by comparing a weighted sum of alarm values to an area's importance threshold. The weight, or believability, of each alarm is based upon the current environmental conditions and the sensor's past performance. The sensor will have a high weight if weather conditions are favorable and its past performance has been reliable based upon the operator's assessments of system alarms.

The weighting function affects a binary alarm state received from the sensor, no analog information from the sensors is being used. The circular area surrounding each RSS sensor pod is divided into twelve equal wedges. The wedge priorities are initialized by the system supervisor when the RSS is initially deployed in an area. Currently, each wedge can be assigned a high, medium or low priority.

Each time an alarm occurs, the weight for the sensor in alarm is computed. The calculation takes into account environmental factors, which affect each sensor differently, and the sensor's performance history, which affects each sensor equally. The weighting functions were determined empirically to achieve the desired performance. Each sensor alarm can have a weight which varies between 0 and 100 and is proportional to the sensor's believability. When the alarm first arrives, this weight is set based upon the environmental conditions. Next a performance offset is subtracted from the environmentally weighted value to account for poor past performance. Three assessments can be made by the system operator: real, nuisance, and unknown. The performance offset is initialized to 0 on system startup, real alarms reduce the offset by 5, nuisance alarms leave it the same, and unknown alarms increase the offset by 5. The system also decays the offset automatically by 1 every 30 seconds so that a sensor can eventually return to its maximum weight under the given environmental conditions.

The VMD can alarm due to the effects of high winds, precipitation or changing light conditions as well as intruders. Each of these conditions is assigned an equal effect on the weight of a VMD alarm. A linear function varies the alarm weight between 0 and 100 for wind speeds of 40mph and 0mph respectively. Precipitation is accounted for by adding 50 to the weight if precipitation is present or 100 if not. Changing light conditions add an additional 50 to the overall weight while unchanging conditions contribute 100 to the value. The total VMD alarm weight is now a value between 100 and 300. This total is divided by 3 to result in a normalized value of 33.3 to 100.

The PIMS, GSR, and Southwest Microwave sensors are affected by wind only. Their environmental weights are linearly varied between 0 and 100 for wind speeds of 40mph and 0mph respectively.

The ADTACS believability is affected by the acoustic noise floor which can result from wind noise or other background noises picked up by the microphones. The ADTACS system sends out a confidence measure with each alarm it generates. Low confidence alarms are given a weight of 50 while high confidence alarms are given 100.

As alarms are received by the OCU, their weighted values are time stamped and stored in a queue. During each iteration of the OCU main loop, the alarms occurring within the past 5 seconds for each particular wedge are summed up and compared with the wedge threshold. High priority wedges are given thresholds of 25, middle priority wedges are given thresholds of 50, and low priority wedges are given thresholds of 80. When the weighted sum of alarms exceeds the wedge threshold, the operator is notified of this system alarm and the alarm queue for that particular wedge is flushed out.

4.3 Robotic Communication Protocol (RCP)

The communication protocol between the OCU and TMSS has been completely overhauled to add error checking capabilities. Initially, MaPSS was the only sensor pod attached to the RSS OCU. Error checking was never required with MaPSS due to the high reliability of the fiber optic data link. Bad data is more likely to be received over TMSS's RF data link, with the probability of occurrence depending upon terrain and transmission distance. Due to safety considerations, it is critical that false commands are not received by a moving robotic vehicle. The basic elements of the new Robotic Communication Protocol are outlined in Appendix B.

5.0 Planned Future Improvements

The RSS program has just over one year of exploratory development remaining before the system transitions to advanced development. Emphasis will be placed on the development of sensor fusion, autonomous navigation, system testing and documentation.

Development of a neural network to process intrusion detection sensors is proceeding. The General Research Corporation (GRC) Automated Signal Processor (ASP) has just been received. The ASP has four serial ports, three 8 bit I/O ports, and will accept a total of 16 single ended or 8 differential inputs to the on-board A/D which can sample signals at up to 25kHz. This processor will provide a flexible system which will allow us to input a variety of signals into the neural network during development.

Training a neural network requires a large amount of ground truth data representing the full spectrum of conditions that will be encountered by the system. An extensive data collection effort has begun in which signals from all of the RSS sensors will be stored on tape for a variety of intrusion and nuisance sources occurring in a wide range of weather conditions with different site backgrounds. This data base will be used to train the ASP and allow us to determine if this approach to sensor fusion will produce valid results.

Current efforts will complete benchtop testing of the ASP using taped data during the summer of 1991. Actual integration of the ASP hardware into the TMSS platform is planned for the fall of 1991 if results from benchtop testing are promising.

The capability will be developed to allow TMSS to autonomously navigate in structured exterior environments and detect obstacles in its path. Path following will be accomplished using a combination of information from on-board dead reckoning sensors and an external position location beacon. Simple obstacle detection using ultrasonic sensors will enable TMSS to detect obstacles in its path, stop, and notify the operator. The operator would then be required to teleoperate the robot past the obstacle before resuming autonomous operations. The year end demonstration of this capability will consist of TMSS traversing a preplanned security patrol path and executing intrusion detection functions at designated surveillance points along the path. Obstacle detection will also be demonstrated during this fixed site patrol.

Extensive system performance testing will be conducted at Sandia to determine detection patterns, probability of detection (POD), and nuisance alarm rates (NAR).

Complete system documentation addressing system operation and capabilities, as well as hardware and software documentation will be provided at the end of exploratory development.

Appendix A

Communication Protocol

for the

6805 Control Processors on TMSS and MaPSS

This report documents the commands and data formats that are used to communicate with the 6805 control microprocessors used on TMSS and MaPSS.

Commands to the MaPSS 6805 processor are sent over a serial port that is set up for 9600 baud, 8 data bits, 1 stop bit, no parity, CLS and DSR disabled. The TMSS 6805 processor uses an identical serial port set up for communication at 1200 baud, 8 data bits, 1 stop bit, no parity, CLS and DSR disabled. Both TMSS and MaPSS communicate through COM1, TMSS has an additional COM2 port that is not currently used.

Binary commands are sent to the 6805 along with optional data bytes that are sent in HEX/ASCII format. Legal command values are 1-47, 58-64, and 71-255; in other words, any value except 0-9 and A-F ASCII. Currently these commands are limited to 1-38 decimal. The data bytes that may precede command bytes are sent in HEX/ ASCII format, note that A - F mu be capital letters.

The following is a list of valid commands for the 6805 along with the required data bytes and their units. Any response to the command is also noted. Decimal values are represented strictly as numbers, Hex values are preceded by a $.

Command 01: Read from memory location. (^A)

Command is valid for both TMSS and MaPSS. Binary command byte (01) is preceded by 4 HEX/ASCII characters representing the specific address you would like to read. eg. Hi Address/Lo Address/01. Note that the Hi and Lo addresses are actually sent over as two ASCII characters each.

Command 02: Write to memory location. (^B)

Command is valid for both TMSS and MaPSS. Binary command byte (02) is preceded by 6 HEX/ASCII characters representing the specific address and data byte you would like to write. eg. Hi Address/Lo Address/Data Byte/02. Note that the Hi and Lo addresses and the Data byte are actually sent over as two ASCII characters each.

Command 03: Input a new Steering Value. (^C)

This command is valid for TMSS only. The command is preceded by 2 ASCII characters representing the desired steering position. Full left = $FF, Full right = $00, Center = $80.

Command 04: Tilt down (^D)

Command is valid for both TMSS and MaPSS. No data bytes are required. Continue action until tilt stop is received.

Command 05: Zoom Stop. (^E)

Command is valid for both TMSS and MaPSS. No data bytes are required.

Command 06: Focus in. (^F)

Command is valid for both TMSS and MaPSS. No data bytes are required. Continue action until focus stop is received.

Command 07: Focus out. (^G)

Command is valid for both TMSS and MaPSS. No data bytes are required. Continue action until focus stop is received.

Command 08: Focus stop. (^H)

Command is valid for both TMSS and MaPSS. No data bytes are required.

Command 09: Zoom in. (^I)

Command is valid for both TMSS and MaPSS. No data bytes are required. Continue action until zoom stop is received.

Command 10: Turn spotlight on. (^J)

Command is valid for both TMSS and MaPSS. No data bytes are required. Continue action until spotlight off is received.

Command 11: Turn Spotlight off. (^K)

Command is valid for both TMSS and MaPSS. No data bytes are required. This is the default state.

Command 12: Pan Left. (^L)

Command is valid for both TMSS and MaPSS. No data bytes are required. Continue action until pan stop is received. Note, this pan unit corresponds to RCP pan unit # 0.

Command 13: Input a new Brake Throttle Value (^M)

This command is valid for TMSS only. The command is preceded by 2 ASCII characters representing the desired brake/throttle position. Full brake = $B1, Full Throttle = $5B, Null = $80.

Command 14: Input new Kill word. (^N)

This command is valid for TMSS only. The command is preceded by 1 ASCII character which represents the state of the choke, ignition and start bits. Note that the ignition bit is synonymous with the software kill on TMSS.
Bit 0 = Start bit, "1" turns on the starter.
Bit 1 = Ignition bit, "1" turns on the ignition, "0" activates the software kill.
Bit 2 = Choke bit, "1" turns on the choke.
Bit 3 = Not used.

Command 15: Zoom out. (^O)

Command is valid for both TMSS and MaPSS. No data bytes are required. Continue action until zoom stop is received.

Command 16: Pan to absolute position. (^P)

Command is valid for both TMSS and MaPSS. This pan unit corresponds to pan #0 in RCP. Execute closed loop control to the pan location indicated in the previous 3 ASCII characters (only 10 valid bits). Full left = $000, full right = $3FF.

Command 17: Request for vehicle status. (^Q)

Command is valid for TMSS only. No data bytes are required. TMSS sends the following information in response to this command.

Byte 0: Actual Steering position.

Byte 1: Actual Brake/Throttle Position

Byte 2: Bit 0 is Hi Compass bit.

Bits 4,5,&6 are the gear counter.
0 = Reverse,
1 = Neutral,
2 = First,
3 = Second,
4 = Third,
5 = Fourth,
6 = Fifth.
Bit 7 is Nod mode ("1" indicates steering slaved mode)

Byte 3: Lo Compass byte. This byte is combined with the compass bit from byte 2 to obtain compass angle. Value will range 0 - 359 in units of degrees.

Byte 4: Incremental Odometer. The change in odometer value since the last request. Range is 0 - 255 in units of 1.9311".

Byte 5: Actual Camera Nod Position. Range is 0 - 255 with full left = $00 and full right = $FF.

Byte 6: Engine Speed. Range is 0 - 255, units are 140.515 RPM.

Byte 7: Vehicle Speed. Range is 0 - 255, units are 4.522 inches per second.

Byte 8: Vehicle Pitch angle. Range is 0 - 255, Full pitch back = $00, Level = $80, Full pitch forward = $FF. Units are 0.52 degrees.

Byte 9: Vehicle Roll Angle. Range is 0 - 255, Full roll right(left side up) = $00, Level = $80, Full roll left (right side up) = $FF. Units are 0.52 degrees.

Command 18: Pan Right. (^R)

Command is valid for both TMSS and MaPSS. No data bytes are required. Continue action until pan stop is received. Note, this pan unit corresponds to RCP pan unit # 0.

Command 19: Request for alarm status. (^S)

Command is valid for both TMSS and MaPSS. No data bytes are required. The following information is sent in response to this command.

Byte 0: Pan #0 angle high byte. Only the two low bits are valid, total pan word is 10 bits long. Combine the 2 LSB of byte 0 with the entire 8 bits of byte 1 to obtain the pan value. Full left pan = $000, Full right pan = $3FF. Units are 0.352 degrees. Range is 0 deg to 359 deg.

Byte 1: Pan #0 angle low byte. See byte 0 for a description of this data.

Byte 2: Tilt #0 angle. This angle is contained in one 8 bit word. Full tilt down = $FF, Full tilt up = $00. Units are 0.352 degrees, Range is -45 deg to + 45 deg.

Byte 3: Alarm word.
  Bit 0: PIMS Alarm (1 = Alarm)
  Bit 1: Microwave Alarm (1 = Alarm)
  Bit 2: Precipitation Detector (1 = Precip)
    Note: Bit 2 = 0 for TMSS at all times.
  Bits 3-7: No data.

Byte 4: Solar data. Range is $00 - $FF, with no scale. Complete darkness = $00, Full sun = $FF. This byte is always 0 for TMSS.

Byte 5: Ambient temperature. Value = 79.36 + 1.781(degrees C). The range is $00 - $FF and the units are 0.561 degrees per state. This byte is always 0 for TMSS.

Byte 6: Wind speed. Value = 1.969(mph) and the units are 0.51 mph per state. This byte is always 0 for TMSS.

Command 20: Tilt to absolute position. (^T)

Command is valid for both TMSS and MaPSS. This tilt unit corresponds to tilt #0 in RCP. Execute closed loop control to the tilt location indicated in the previous 2 ASCII characters (8 bits). Full down = $FF, full up = $00.

Command 21: Tilt Up. (^U)

Command is valid for both TMSS and MaPSS. No data bytes are required. Continue action until tilt stop is received.

Command 22: Shift Up. (^V)

Command is valid for TMSS only. No data bytes are required. Initiates a shift to the next highest gear if possible.

Command 23: Shift Down. (^W)

Command is valid for TMSS only. No data bytes are required. Initiates a shift to the next lowest gear if possible.

Command 24: Pan Stop. (^X)

Command is valid for TMSS and MaPSS. No data bytes are required. Stops movement of pan #0, used in conjunction with manual pan commands.

Command 25: Tilt Stop. (^Y)

Command is valid for TMSS and MaPSS. No data bytes are required. Stops movement of tilt #0, used in conjunction with manual tilt commands.

Command 26: Nod Right. (^Z)

Command is valid for TMSS only. No data bytes are required. Initiates pan left motion for pan #1 (nod unit).

Command 27: Nod Left.

Command is valid for TMSS only. No data bytes are required. Initiates pan right motion for pan #1 (nod unit).

Command 28: Nod Stop.

Command is valid for TMSS only. No data bytes are required. Stops pan motion for pan #1 (nod unit).

Command 29: Nod to Absolute Position. (Not currently implemented.)

Command is valid for TMSS only. This pan unit corresponds to pan #1 in RCP. Execute closed loop control to the pan location indicated in the previous 2 ASCII characters (8 bits).
Full left = $00, full right = $FF.

Command 30: Mast Up.

Command is valid for TMSS only. Raises the pneumatic mast.

Command 31: Mast Down.

Command is valid for TMSS only. Lowers the pneumatic mast.

Command 32: Mast Stop.

Command is valid for TMSS only. Stops movement of the pneumatic mast and holds its position.

Command 33: Initiate Steering Slave Nod Mode.

Command is valid for TMSS only. Places the camera nod unit (pan #1) in steering slave mode, so that the nod direction matches the steering direction. The directions are reversed if TMSS is in reverse gear. If this mode is activated all manual nod commands are ignored.

Command 34: Initiate Manual Nod Mode.

Command is valid for TMSS only. Places the camera nod unit (pan #1) in manual mode, so that the nod unit obeys manual commands only. This is the default mode at startup.

Command 35: Turn on Video Transmitter.

Applies power to the video transmitter, the power level must be set manually with the switch on the unit. Default state is power on.

Command 36: Turn off Video Transmitter.

Disconnects power from the video transmitter.

Command 37: Turn on Sensor Pod Power.

Applies power to the video camera and the intrusion detection sensors. Default state is on.

Command 38: Turn off Sensor Pod Power.

Disconnects power to the video camera and intrusion detection sensors.

Appendix B

Robotic Communication Protocol (RCP)

3/21/91

1.0 Introduction

Standardization of a communication protocol for robotic vehicles is desirable to reduce developmental efforts for new systems. The need for generating new communications software or building a new driving station for each new project could be eliminated. The standardized protocol must be designed so that it is capable of performing all of the functions that are currently required for robotic vehicle control and must be expandable so that unforseen capabilities may be added at a later date. Flexibility is generally in direct conflict with a protocol's efficiency, but the limited bandwidth available for RF communication links dictates that an attempt be made to make the protocol as efficient as possible.

This document defines the Robotic Communication Protocol (RCP) developed at Sandia National Laboratories to meet the above requirements. The RCP addresses the middle layer of the entire communication system and is responsible for defining how information is encoded into messages. The lowest layer of the communication system is the hardware layer. This layer is in charge of sending and receiving the actual bytes of data over the communication media. The highest layer of the communication system is the application layer. This layer is responsible for the interface between the RCP and the remainder of the robotic control system.

In addition to the interface with the communication media, the hardware layer is responsible for adding data encryption and forward error correction to the RCP messages if required by the application. These functions are generally available in hardware, so were left for implementation in this lowest level. The RS-232 serial communication protocol has been chosen for use in the hardware layer for the robotic vehicle projects at Sandia. Other hardware layer communication protocols could be chosen depending upon application needs.

The application layer manages the flow of information sent over the communication link and the interface between the RCP data structures and the application program. This layer is also responsible for directing communications among the individual members of the communication network.

2.0 Network Definition

Certain applications for robotic vehicle systems have a need to control multiple robots from a single control console. This may require that multiple robots communicate over a single RF communication channel. In these instances there must be some method of arbitration to determine who has control of the communication link at any given time.

The RCP assumes the availability of full duplex data links for optimal performance. Full duplex systems allow communications to occur in both directions simultaneously without interference. In half duplex or simplex systems, communications can only occur in one direction at a time and the RCP will be restricted to operating in query mode. More detail about the two RCP communication modes, query and stream, will be discussed later in this report.

Each communication link is assigned one master, who will have exclusive control of the outgoing transmit frequency, and one or more slaves that will be given control of the incoming transmit frequency in turn. It is the duty of the communication link master to arbitrate control of the return data link among the slaves. In general, this link master will be a Control Driving Station, or CDS, and the link slaves will be Robotic Vehicles, or RV's. This notation will be used in the remainder of the report when referring to the link master and slaves.

The communication network must be defined in the application software. This involves defining the connectivity matrix for that network and a link master. The link master is responsible for passing a token to the robot allowed control over the return link; the master always has control of the outgoing link.

Individuals in the network are assigned a unique ID number between 0 and 126 for the purpose of directing query mode commands and for passing the token to the sender of streamed mode data. The ID number 127 is used as a universal ID to access all robots. A query mode command sent to 127 is intended for all robots. In the case that the token is passed to 127, all robots can talk. This feature is intended to be used in a quiet mode only, where the robots are directed to wake up and send a burst of data upon an alarm condition. In the event that two robots wake up at the same time and interference occurs, the master will query the individual robots for data.

To illustrate the system described above, refer to the communication network shown in figure 1 depicting the Fire Ant concept developed at Sandia. This network consists of three separate communication links with a primary and secondary layer of robots which are to be controlled from the CDS. In this example, the CDS controls the two Queen Ants directly and the five Fire Ants indirectly through their respective Queen Ants.

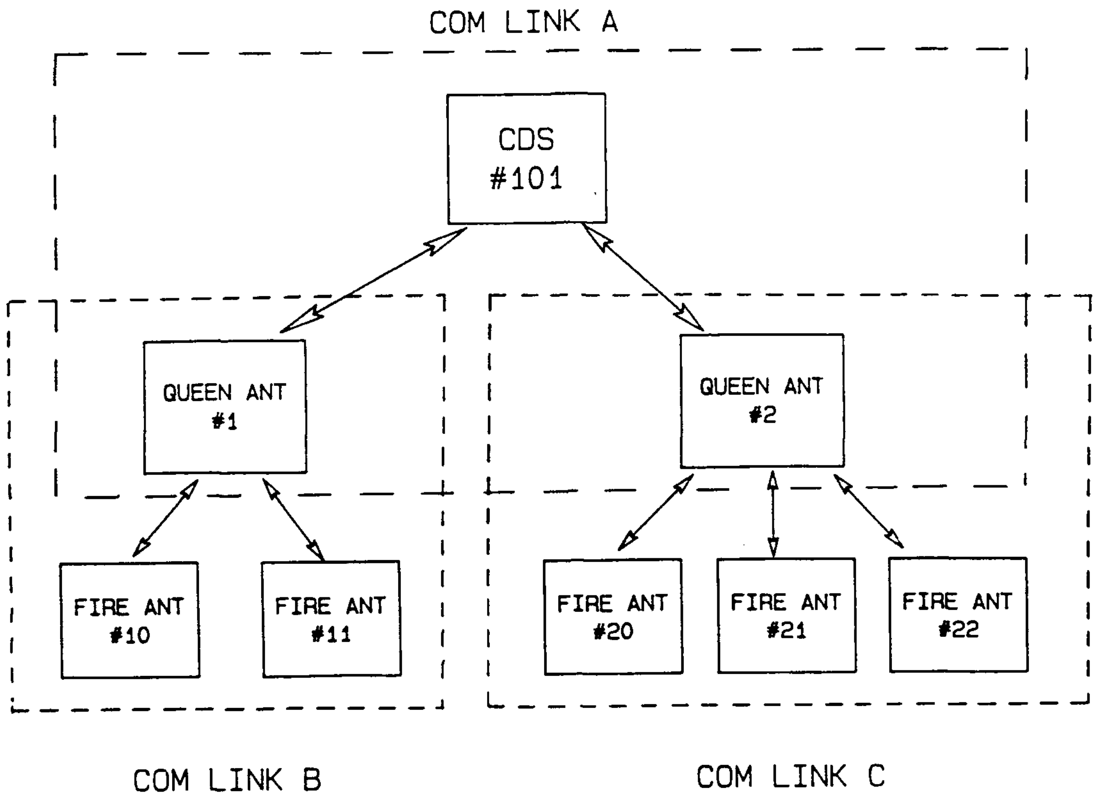

Figure 1. Example Communication Network

In the above example, the Link A master is number 101, the Link B master is number 1, and the Link C master is number 2. The system must have the network connectivity defined to specify the proper paths for communication. The communication paths for the example network are listed in Table 1 below. For example, CDS 101 can send a message to Fire Ant 21 knowing that the message must pass through Queen Ant 2 on link A. Queen Ant 2 will then pass this message on to Fire Ant 21 over link C. Management of the information flow within the network is the responsibility of the software in the application layer and is not explicitly handled by the RCP.

Table 1
Communication Paths for Example Network

Link A	Link B	Link C
101/1	1/101	2/101
101/2	1/10	2/20
101/1/10	1/11	2/21
101/1/11	1/127	2/22
101/2/20		2/127
101/2/21		
101/2/22		
101/127		

3.0 Message Structure

The string of bytes which make up an RCP message, or data block, can be sub-divided into "command" and "data" bytes. There will be one command byte per string and multiple data bytes. All data bytes will be identified by the high bit (bit 7) being set, while command bytes will have this bit cleared. Thus a data byte will appear as 1xxx xxxx, while a command byte will appear as 0xxx xxxx. The x's in the above definitions are bits which can be either set or cleared depending on the information in the byte.

The format of a data block will be a number of data bytes followed by a checksum byte and terminated with one command byte. Figure 2 below shows the construction of an RCP data block.

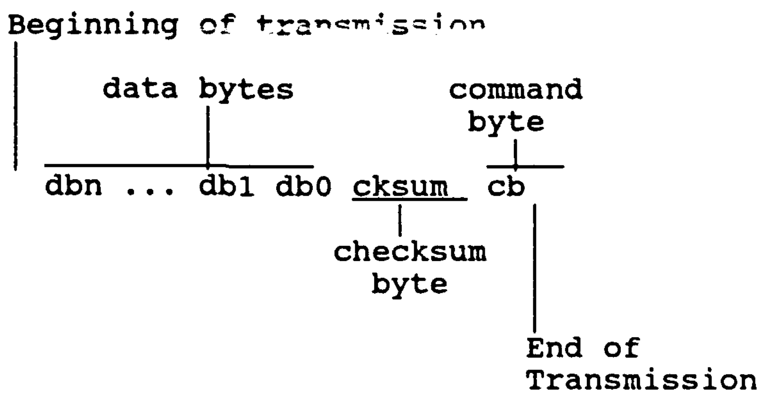

Figure 2. Construction of an RCP data block

4.0 Error Checking

Due to the potentially dangerous nature of remotely controlling a mobile robot, error checking was thought to be a necessary component of the RCP. In past experience, the data links used for robotic vehicle control at Sandia are generally robust and

very few bad data bytes are observed. In light of this experience, a simple checksum scheme is used. More elaborate error checking algorithms could have been chosen at the expense of greatly increased overhead. Any application requiring additional robustness in the communication link can implement forward error correcting at the hardware layer.

The checksum byte is computed by summing the values of all data bytes and the command byte and then adding the number of bytes in the data block to the result. The checksum byte is represented as a data byte with the high bit set. The checksum contains only seven bits of information, all carry bits are thrown away.

As an example of calculating a checksum byte, assume the data block looks like the following:

(1110 1000, 1111 1111, cksum, 0000 0010)

The checksum is computed as follows:

```
  110 1000 - value of 1st data byte
  111 1111 - value of 2nd data byte
  000 0010 - value of command byte
 +000 0100 - total # of bytes in block
1 110 1101 - resulting checksum
```

Note that only the low seven bits of information are used in calculating the checksum and only seven bits are valid in the result. Make sure the high bit of the resulting checksum byte is set, so that it looks like a data byte.

The resulting checksum byte is 1110 1101.

4.0 Communication Modes

The are two basic modes of communication supported with the RCP. The first is stream mode in which the CDS and a chosen RV continuously send predefined data messages over the link. Both sides of the communication link are talking at the same time in stream mode. The second is query mode in which the CDS sends one message to a specific RV on the link and waits for a response. Only one side of the communication link is allowed to talk at a time in query mode.

4.1 Stream Mode

The stream communication mode is used to efficiently send continuous updates of important data. Some examples of when this mode is desirable are sending steering actuator positions to a robot during teleoperation or compass data to the control console for off-board navigation calculations. During these types of

operations it is necessary to update certain pieces of information, such as steering angle, at a fast rate while other pieces of information, such as gear position, are not as critical. Stream mode allows the user to build up predefined blocks of data which are to be sent over the communication link continuously, without the overhead associated with queries.

As an example of how this mode is intended to be used, suppose that a robot is being teleoperated and dead reckoning position estimates of the robot's position are being calculated at the CDS. The robot requires continuous updates of steering, brake and throttle actuator positions and occasional updates of camera, and gear shifter controls. The CDS requires constant updates of compass and odometer information and occasional updates of pitch and roll angles. Define the following packets of information:

P0: Steering actuator position, 7 bits,
P2: Throttle actuator position, 7 bits,
P4: Brake actuator position, 7 bits,
P8: Transmission gear, 3 bits,
P22: Pitch value, 7 bits,
P23: Roll value, 7 bits.
P32: Driving camera controls, 4 bits,
P44: Compass heading, 10 bits,
P48: Absolute odometer reading, 16 bits,

The control console application software builds the following blocks of data from the list of available data packets.

B0: P8,P0,P2,P4
B1: P32,P0,P2,P4
B2: P22,P44,P48
B3: P23,P44,P48

An example of an actual data block is given below.

B0: Control Block 0, sent by CDS

```
1p8p8p8p0p0p0p0 / 1p0p0p0p2p2p2p2 / 1p2p2p2p4p4p4p4
/1p4p4p40000/ 1$$$$$$$ / 0xxxxxxx
```

where,

```
1  = a leading 1, indicating a data or checksum byte
0  = a leading 0, indicating a command byte
p0 = the 7 bits of packet 0
p2 = the 7 bits of packet 2
p4 = the 7 bits of packet 4
p8 = the 3 bits of packet 8
0  = unused, 0 filled bits
$  = the 7 bits of the checksum byte
x  = the 7 bits of the command byte
```

Note some characteristics of this construction. The spaces and slashes in the above description are not actually sent, but are included for clarity to delineate byte divisions. All data bytes including the checksum byte contain 7 bits of data preceded by a 1. Only the command byte is preceded with a leading 0. The packets are ordered in the sequence that they were constructed. The bits from the individual packets are filled into data bytes with no unused bits until possibly the last byte in the block. Any unused bits are then filled with zeros.

In order to transfer data efficiently, the control console specifies that a continuous data stream will be sent to the robot that alternates between B0 and B1. Similarly, the robot is requested to send a continuous data stream which alternates between B2 and B3. In this manner, the less important pieces of information contained in packets P8, P22, P23, and P32 are interleaved with the more important information sent in every data block. The overhead associated with a request for specific information is eliminated since the set up occurs previous to the initiation of the data stream. Any errors that occur during transmission result in the entire block being ignored. Since important data is sent continuously, it is not necessary for the recipient to request retransmission of blocks containing errors.

The RCP allows sending 64 possible data blocks which are specifically defined in each application. Command byte values of 0 through 63 indicate that the corresponding previously defined block has just been sent; while values of 64 through 127 are reserved for discreet commands such as "build block", or "kill robot". The content of a specific block is application dependent; one application may use a certain set of block definitions while another may use completely different ones. Data blocks are defined in software header files or at system startup through a handshaking technique described in the section on initialization.

4.2 Query Mode

In some applications, the robotic vehicles may not require the fast update rates that stream mode operations are intended to provide. Instead, the CDS may be acting as more of a supervisor of autonomous robots in which the intensive computing is done on-board. These situations are better suited to query mode communications, in which the CDS is sequentially exchanging information with each RV connected to the link. In query mode, the CDS will send a command to a specific robot. This robot will then act on the command or return the requested data. Unlike stream mode, the robot relinquishes control of the return communication link after sending the requested information.

5.0 Initialization

In order to enhance the efficiency of the RCP while retaining its flexibility, some a priori knowledge of the information to be sent over the communication link was assumed. Thus, the block contents for a specific application are defined in an initialization procedure. The information contained in a specific block is then inherently known by the application without the overhead associated with specifically defining each data packet every time that it is sent over the link.

There are three commands defined in the RCP that allow the CDS to configure the data blocks during system startup. This feature allows the CDS to control robots using the RCP that were not specifically designed for that particular system. The initialization commands allow the CDS to define a data block, define a data stream and allow a robot to respond that a certain command or data packet is not supported. Refer to the appendix discussing the command set for the specific details on these commands.

As an example of this initialization sequence, suppose that a CDS having the capability to control a robot with an on board navigation system is being used to operate a vehicle that has a teleoperation capability only. The CDS has been predefined as ID number 101 and the RV as ID number 1. Upon system startup, CDS101 sends a "define block 0" command to RV1 indicating that block 0 contains packets for steering, brake and throttle commands. Initially RV1 does not respond, since it has not been powered up, and CDS101 continues to send the command waiting for a response. Upon power up, RV1 echoes the command indicating that it has received and acknowledged the data block. CDS101 then sends a command defining block 1 to contain the robot's X and Y position status. Since RV1 has no navigation capabilities, it responds with a command indicating no support of the position status packets. CDS101 now knows that this capability is not available from RV1 and goes on to define other blocks necessary for teleoperation. Once the blocks have been defined, CDS 101 defines the stream of blocks to and from RV1, selects RV1 for stream mode and begins communications using the predefined data streams.

6.0 Implementation State

A trial implementation of the RCP is being tested at Sandia on the Remote Security Station (RSS) project. The RSS system consists of both stationary and mobile robotic security platforms. Separate communication links are used to control each of the sensor systems. The RSS system is being modified to use the RCP for control of the mobile robotic sensor system only.

This initial implementation is intended to test the basic features of the protocol. Use of the RCP has been an improvement over the original protocol used for the RSS. The advantages include the capability for error checking and more efficient use of the communication bandwidth.

Features of the RCP that have not been tested in this initial implementation are stream mode communications, multiple vehicle control, and run time initialization of data blocks. Stream mode communications were not implemented on the RSS project because the original control console logic was not set up to operate in this manner. Instead, the query mode was implemented to eliminate major rework of the RSS CDS software. Multiple vehicle control has not been tested due to the lack of an appropriate application. Run time data block initialization is an advanced feature of RCP that is necessary if true interoperability of hardware is to be realized. The code to implement this feature has been put off until the basic features of the protocol have been fully tested. Block definitions are currently being implemented via configuration files that must be present when the communication software is compiled. In the research atmosphere that is present at Sandia, the definition of data blocks through configuration files is an acceptable approach that greatly simplifies the code necessary for implementation of the RCP.

The command and packet sets given in the appendices were defined so that any of the functions required by current robotic programs at Sandia could be implemented. This is certainly not an exhaustive list and is intended to grow as new functions are required. The packet numbers are approaching the original limit of 128 and will most certainly need to be extended.

7.0 Definitions and Abbreviations

CDS - Control Driving Station, used to indicate a communication link master.

Checksum Byte - The final data byte in a block, this byte contains information that is used to verify that no errors have occurred during transmission of a block.

Command Byte - Any byte with a leading zero, the command byte is used to indicate how the preceding data should be interpreted.

Data Block - One or more RCP data packets that are packaged together with checksum and command bytes and sent over the communication link as a group.

Data Byte - Any byte with a leading one, data bytes consist of data packets and checksum information. The content of any data byte is defined by its associated command byte.

Data Packet - The smallest division of RCP data, such as a steering angle command.

Data Stream - A sequence of data blocks that are continuously sent over the communication link in stream mode.

Query Mode - A communication mode in which only one side of the communication link is talking at any time.

RCP - Robotic Communication Protocol.

RV - Robotic Vehicle, used to indicate a communication link slave.

Stream Mode - A communication mode in which predefined streams of data are continually being sent over the communication link in both directions. Both sides of the communication link are talking simultaneously.

Appendix C

RCP Command Set

The commands listed in this appendix apply to the Robotic Communication Protocol (RCP) introduced in Appendix B.

In the following definitions, the communication link master will be assumed to be a control driving station and will be referred to as the CDS. The link slave will be assumed to be a robotic vehicle and be designated as the RV. These assumptions may not always apply, but this terminology is used for simplicity.

Some of the commands that will be defined have a specific source and recipient identified in the command. In others, this is implicitly known due to some previous setup. There occasionally may be instances that a command is being relayed by some intermediate node in the network. The relay vehicle must assume the role of the source or recipient ID as necessary to make the transaction appear transparent to the actual source and recipient of the command.

Commands 0 - 63 are commands that are reserved to designate which previously defined block is currently being sent. Commands 64 - 127 are reserved to initialize stream mode operations and perform other discreet functions.

Commands 0 - 63 are of the following form:

Sending Data Block #

Form: (dbn...db1 cksum cb) sent by CDS or RV

dbn = The first data byte sent in the current data block

db1 = The last data byte sent in the current data block.

cb = decimal value of the data block being sent, range 0 - 63.

Response: None

64. Define Data Block

Form: (dbn... db4 db3 db2 db1 cksum cb) sent by CDS or RV

dbn - db4 = Data bytes defining the packet numbers to be sent in this block. These numbers currently range from 0 to 127, but can be expanded later if additional packets are needed. The block is constructed in the order that the packet numbers are listed in the definition. When the defined data block is actually sent, the packet designated by dbn is sent first and db4 is sent last, followed only by the checksum and command bytes.

db3= The number of the data block being defined. This number can range from 0 to 63 and takes up one entire data byte.

db2= The ID# of the data block's destination

db1= The ID# of the data block's source

cb = decimal 64.

Response: The command recipient echoes the command exactly if all packets are supported. In the case that some packet in the block is not supported by the recipient, it will respond with command 67 indicating no support of the specified packet.

65. Select/Deselect an RV for Stream Mode

Form: (db3 db2 db1 cksum cb) sent by CDS only

db3 = select/deselect flag
- 0 = select RV for query mode
- 1 = select RV for stream mode
- 2 = select RV for automatic wake up on alarm, see quiet and surveillance modes in packets P96 and P97.

db2 = ID of stream source
db1 = ID of stream destination
cb = decimal 65

Note: Each instance of this command is used to activate the stream mode communications in one direction only. This allows distinct RV's to be selected to receive and send data streams. The command must be issued twice for selecting a single RV to both send and receive data streams.

Response: Selected RV echoes the command in identical form

66. Define data block sequence for stream mode.

Form: (dbn ... db3 db2 db1 cksum cb) sent by CDS or RV

dbn - db3 = order of previously defined data blocks to be sent in stream mode. Range is 0 - 63, each data block number occupies an entire data byte.

db2 = ID of stream source

db1 = ID of stream recipient

cb = decimal 66

Response: Command recipient echoes the command in identical form.

67. No Support of Command/Packet.

Form: (dbn ... db2 db1 cksum cb) sent by RV

dbn - db2 = list of command or packet types not supported by the RV. Note that these must be of the same type, commands or packets within each instance. Each command or packet number occupies an entire byte.

db1 = Command/Packet listing flag,
if db1 = 0, dbn-db2 is a list of commands
if db1 = 1, dbn-db2 is a list of packets.

cb = decimal 67

Response: None.

68. Repeat Message.

Form: (db2 db1 cksum cb) sent by CDS or RV

db2: ID number of the requester.

db1: ID number of the recipient.

cb = decimal 68

Response: The recipient defined in db1 repeats the last message that it sent to the requester defined in db2.

69. Resume Communications. (XON)

Form: (cksum cb) sent by CDS or RV

cb = decimal 69.

Response: The recipient will resume normal communications that were previously halted by an XOFF command.

70. Suspend Communications. (XOFF)

Form: (cksum cb) sent by CDS or RV

cb = decimal 70.

Response: The recipient will halt communications at the current point to be resumed after the receipt of an XON.

71. Query for Data Block.

Form: (db3 db2 db1 chsum cb) sent by CDS or RV

db3: ID number of the requester

db2: ID of RV or CDS being queried

db1: Number of block to send back

cb = decimal 71

Response: The ID defined in db2 sends the requested data block using the appropriate command from commands 0 - 63.

Appendix D

RCP Packet Definitions

The packets listed in this appendix apply to the Robotic Communication Protocol (RCP) introduced in Appendix A.

The RCP packet definitions are given below. Any data that will conceivably be sent in both directions is defined as two packets, one for command data and one for status data. This conveniently provides RCP with separate data locations for incoming and outgoing data. Command packets will conventionally be reserved for data being sent from the CDS to the robot and status packets for data in the opposite direction, although alternate definitions could be used for specific applications.

P0 Steering Angle Command
P1 Steering Angle Status
Packet Definitions: packet length is 7 bits, value is 0 - 127 inclusive; 0 = full left, 64 = center, 127 = full right.

P2 Throttle Value Command
P3 Throttle Value Status
Packet Definitions: packet length is 7 bits, value is 0 - 127 inclusive; 0 = no throttle, 127 = full throttle.

P4 Brake Value Command
P5 Brake Value Status
Packet Definitions: packet length is 7 bits, value is 0 - 127 inclusive; 0 = no brake, 127 = full brake.

P6 Engine RPM Status
Packet Definition: packet length is 7 bits, value is 0 - 127 inclusive and represents *(RPM/100)*; 0 = 0 RPM, 127 = 12,700 RPM.

P7 Oil Pressure Status
Packet Definition: packet length is 7 bits, value is 0 - 127 inclusive and units are (lbs/sq in); 0 = 0 lbs/sq in, 127 = 127 lbs/sq in.

P8 Transmission Gear Command
P9 Transmission Gear Status
Packet Definitions: packet length is 3 bits, value is 0 - 7 inclusive;

0 = Park		
1 = Reverse		
2 = Neutral		
3 = First		Drive 1
4 = Second	or	Drive 2
5 = Third		Drive
6 = Fourth		
7 = Not in Gear		

P10 Transfer Case Gear Command
P11 Transfer Case Gear Status
Packet Definitions: packet length is 2 bits, value is 0 - 3 inclusive;

0 = 2WD		0 = High Lock
1 = 4WD High	or for	1 = High
2 = Neutral	full time 4WD:	2 = Neutral
3 = 4WD Low		3 = Low
4 = Not in Gear		4 = Not in Gear

P12 Ignition State Command
P13 Ignition State Status
Packet Definitions: packet length is 1 bit; 0 = ignition off, 1 = ignition on.

P14 Starter State Command
P15 Starter State Status
Packet Definitions: packet length is 1 bit; 0 = starter off, 1 = starter on.

P16 Choke State Command
P17 Choke State Status
Packet Definitions: packet length is 1 bit; 0 = choke off, 1 = choke on.

P18 Light Unit 0 State Command
P19 Light Unit 0 State Status
P20 Light Unit 1 State Command
P21 Light Unit 1 State Status
Packet Definitions: packet length is 1 bit; 0 = off, 1 = on

P22 Pitch Inclination Status
Packet Definition: packet length is 7 bits, range is 0 - 127 inclusive; each count is worth one degree.

0 = full pitch forward, nose down, -64 deg;
64 = level;
127 = full pitch back, nose up, +63 deg.

P23 Roll Inclination Status
Packet Definition: packet length is 7 bits, range is 0 - 127 inclusive; each count is worth one degree

0 = full roll right, left side up, -64 deg
64 = level;
127 = full roll left, right side up, +63 deg.

P24 Pan Unit 0 Command
P25 Pan Unit 0 Status
P26 Pan Unit 1 Command
P27 Pan Unit 1 Status

Packet Definitions: packet length is 17 bits, supports both absolute position and variable rate pan modes.

b16 - b5: 12 bit value representing 0 - 360 degrees, resolution is 0.087890625 degrees per state. 0 = 0 degrees, 4095 = 359.91 degrees.

b4 - b0: 5 bit value representing pan rate, note that this is not based on absolute units but is a variable based on the capabilities of the pan unit. 0 = stopped, 31 = full pan rate.

Note: Pure rate control is accomplished by setting the position to an extreme (eg. 0 or 4095) and setting the rate to the desired value.

P28 Tilt Unit 0 Command
P29 Tilt Unit 0 Status
P30 Tilt Unit 1 Command
P31 Tilt Unit 1 Status

Packet Definitions: packet length is 16 bits, supports both absolute position and variable rate tilt modes.

b15 - b5: 11 bit value representing -90 to +89.9 degrees, resolution is 0.087890625 degrees per state. 0 = -90 degrees, 1024 = 0 degrees, 2047 = +89.91 degrees.

b4 - b0: 5 bit value representing tilt rate, note that this is not based on absolute units but is a variable based on the capabilities of the tilt unit. 0 = stopped, 31 = full tilt rate.

P32 Camera Unit 0 Control Command
P33 Camera Unit 0 Control Status
P34 Camera Unit 1 Control Command
P35 Camera Unit 1 Control Status

Packet Definitions: packet length is 4 bits, controls zoom lens functions.

b3: focus out, 0 = false, 1 = true,
b2: focus in, 0 = false, 1 = true,
b1: zoom out, 0 = false, 1 = true,
b0: zoom in, 0 = false, 1 = true.

P36 Video Transmitter Unit 0 Command
P37 Video Transmitter Unit 0 Status
P38 Video Transmitter Unit 1 Command
P39 Video Transmitter Unit 1 Status

Packet Definitions: packet length is 2 bits, supports 4 power levels on each transmitter, range is 0 - 3; 0 = off, 1 = low power, 2 = med power, 3 = hi power.

P40 Mast Control Command
P41 Mast Control Status
Packet Definitions: packet length is 2 bits.

b1 = mast down, 0 = false, 1 = true
b0 = mast up, 0 = false, 1 = true

P42 Mast Height Command
P43 Mast Height Status
Packet Definitions: packet length is seven bits, range is 0 - 127 inclusive. Each increment in the value represents an increase of 2.5 inches, 0 = 0 inches, 127 = 317.5 inches (26 feet 5.5 inches).

P44 Compass Unit 0 Heading Command
P45 Compass Unit 0 Heading Status
P46 Compass Unit 1 Heading Command
P47 Compass Unit 1 Heading Status
Packet Definitions: packet length is 10 bits. The resolution is 0.352 degrees per count. 0 = 0 deg, 1022 = 359.744 deg, 1023 is not used. 0 degrees is magnetic north with values increasing clockwise around the circle.

P48 Absolute Odometer Value
Packet Definition: packet length is 16 bits, value is 0 - 65,535 inclusive and represents distance travelled in odometer ticks since the system was reset. Default tick distance is 0.1 meter unless changed by the odometer scaling factor.

P49 Incremental Odometer Value
Packet Definition: packet length is 7 bits, value is 0 - 127 inclusive and represents distance travelled in odometer ticks since the last request for odometer data. Default tick distance is 0.1 meter unless changed by the odometer scaling factor.

P50 Odometer Scaling Factor Command
P51 Odometer Scaling Factor Status
Packet Definition: packet length is 11 bits, value is 0 - 2047 inclusive and represents the interpretation of tick distance in the absolute and incremental odometer packets. Each increment of the odometer scaling factor is worth 1 millimeter with 0 representing 1 millimeter and 2047 representing 2048 millimeters. Therefore, if an odometer scaling factor of 999 is given each tick of the odometer will represent 1 meter.

P52 Vehicle Speed Command
P53 Vehicle Speed Status
Packet Definition: packet length is 8 bits, value is 0 - 255 inclusive and represents the vehicle's speed in speed scale ticks per second. The default value of the speed scale factor is 0.1 m/s. For an example of how this construct is used, see the definition of the speed scaling factor below.

P54 Speed Scaling Factor Command
P55 Speed Scaling Factor Status
Packet Definition: packet length is 5 bits, value is 0 - 31 inclusive and represents the interpretation of vehicle speed in increments of 0.1 m/s. This scale is offset so that 0 = 0.1 m/s/tick and 31 = 3.2 m/s/tick. The default value of speed scale is ((0.1 meters per second) per vehicle speed tick). For example, if the default value of 0.1 (m/s)/t is assumed and a vehicle speed of 68 ticks is given, the vehicle speed is 6.8 m/s.

P56 Battery Voltage
Packet Definition: packet length is 8 bits, and value is 0 - 255 inclusive. Resolution is 0.125 volts per increment, so that 0 = 0 volts and 255 = 31.875 volts.

P57 Charging Current
Packet Definition: packet length is 12 bits, value is 0 - 4095 inclusive and represents both charging and discharging currents. Resolution is 0.0625 Amps per increment in value. With 0 = -128.0 Amps (discharging), 2048 = 0 Amps, 4095 = +127.9375 Amps (charging).

P58 Fuel Level
Packet Definition: packet length is 4 bits, value is 0 - 15 inclusive and represents fuel level in 1/15's of a full tank. Empty is represented by 0, full is represented by 15.

P59 Computer Temperature
Packet Definition: packet length is 7 bits, value is 0 - 127 inclusive. Each increment is worth one degree C. So, 0 = -27 deg C, 27 = 0 deg C, and 127 = 100 deg C. Therefore, (deg C = value - 27).

P60 Engine Temperature
Packet Definition: packet length is 7 bits, value is 0 - 127 inclusive. Each increment is worth one degree C. 0 = 0 degrees C, 127 = 127 degrees C.

P61 Ambient Temperature
Packet Definition: packet length is 7 bits, value is 0 - 127 inclusive. Each increment is worth one degree C. So, 0 = -27 deg C, 27 = 0 deg C, and 127 = 100 deg C. Therefore, (deg C = value - 27).

P62 Wind Speed
Packet Definition: packet length is 7 bits, value is 0 - 127 inclusive. Each increment is worth 1 Km/H, 0 = 0 Km/H, 127 = 127 Km/H.

P63 Precipitation Presence
Packet Definition: packet length is 1 bit. A "1" indicates the presence of precipitation and a "0" indicates the absence of precipitation.

P64 Data Transmitter Unit 0 Command
P65 Data Transmitter Unit 0 Status
P66 Data Transmitter Unit 1 Command
P67 Data Transmitter Unit 1 Status
Packet Definitions: packet length is 2 bits. Range is 0 - 3, 0 = off, 1 = low power, 2 = med power, 3 = high power.

P68 Altimeter Value
Packet Definition: packet length is 12 bits, range is 0 - 4095 inclusive with one meter resolution. 0 = -100 meters, 100 = 0 meters, 4095 = 3995 meters. So, Altitude = (value - 100) meters.

P69 Turn Rate Value: TBD, packet length is 10 bits.

P70 X Position Command
P71 X Position Status
Packet Definitions: packet length is 19 bits representing vehicle position in 0.1 meter resolution. Range is 0 - 524,287 inclusive representing a maximum distance of 52.4 Km from the reference point. This reference point is defined by the system's application software.

P72 Y Position Command
P73 Y Position Status
Packet Definitions: packet length is 19 bits representing vehicle position in 0.1 meter resolution. Range is 0 - 524,287 inclusive representing a maximum distance of 52.4 Km from the reference point. This reference point is defined by the system's application software.

P74 UTM Coordinate Command
P75 UTM Coordinate Status
Packet Definitions: TBD, packet length is 10 bits

P76 GPS Coordinates
Packet Definition: TBD, packet length is 10 bits

P77 Del Norte Data
Packet Definition: TBD, packet length is 10 bits

P78 Vehicle Kill State Command
P79 Vehicle Kill State Status
Packet Definitions: packet length is 1 bit. Kill the vehicle if bit is set, keep the vehicle alive if bit is cleared.

P80 Solar Radiation Level
Packet Definition: packet length is 6 bits and represents the ambient solar radiation. Range is 0 - 63 inclusive and is in units of KW/sq M.

P81 Digital Motion Sensor 0 Status
P82 Digital Motion Sensor 1 Status
P83 Digital Motion Sensor 2 Status
P84 Digital Motion Sensor 3 Status
Packet Definitions: packet length is 1 bit, bit set indicates the sensor is in alarm.

P85 Analog Motion Sensor 0 Status
P86 Analog Motion Sensor 1 Status
P87 Analog Motion Sensor 2 Status
P88 Analog Motion Sensor 3 Status
Packet Definitions: packet length is 12 bits representing the signal level of the sensor.

P89 Steering Slave/Fixed Camera Mode Command
P90 Steering Slave/Fixed Camera Mode Status
Packet Definitions: Packet length is one bit. Driving camera is steering slaved if bit is set, camera is fixed if bit is cleared.

P91 Parking Brake Command
P92 Parking Brake Status
Packet Definitions: Packet length is one bit. Parking brake is set if bit is set, else parking brake is off.

P93 Water in Fuel Status
Packet Definition: Packet length is one bit. Water is present in fuel if bit is set, else fuel is OK.

P94 Wait to Start Status
Packet Definition: Packet length is one bit. If bit is set, the RV must wait to start. Typically used for vehicles with glow plugs.

P95 Abort Code Status
Packet Definition: Packet length is seven bits. Contains an application specific code as to the reason for a vehicle abort.

P96 Vehicle Mode Command
Packet definition: Packet length is 2 bits.
0 = Teleoperation Mode
1 = Autonomous Mode
2 = Quiet Mode, shut down all systems for low power consumption, listen for wake up into one of the other modes.
3 = Surveillance Mode, power down transmitters, burst transmissions on alert conditions or CDS requests.

P97 Vehicle Mode Status
Packet definition: Packet length is 2 bits.
0 = Teleoperation Mode
1 = Autonomous Mode
2 = Quiet Mode, shut down all systems for low power consumption, listen for wake up into one of the other modes.
3 = Surveillance Mode, power down transmitters, burst transmissions on alert conditions or CDS requests.

P98 Safety Mode Command
Packet Definition: Packet length is 2 bits.
0 = RECOVER to normal operations from one of the other modes.
1 = STOP, applies full brake and disallows all vehicle actuator commands until RECOVER
2 = SAFE, applies full brake, kills engine and disallows **all** commands until a RECOVER, this is a CDS initiated mode.
3 = ABORT, same as SAFE, except that the mode is initiated by the vehicle due to some on board error condition.

P99 Safety Mode Status
Packet Definition: Packet length is 2 bits.
0 = RECOVER to normal operations from one of the other modes.
1 = STOP, applies full brake and disallows all vehicle actuator commands until RECOVER
2 = SAFE, applies full brake, kills engine and disallows **all** commands until a RECOVER, this is a CDS initiated mode.
3 = ABORT, same as SAFE, except that the mode is initiated by the vehicle due to some on board error condition.

P100 Obstacle Detection ON/OFF Command
Packet Definition: Packet length is 1 bit. Bit cleared indicates that obstacle detection should be turned off and bit set indicates that obstacle detection should be turned on.

P101 Obstacle Detection ON/OFF Status
Packet Definition: Packet length is 1 bit. Bit cleared indicates that obstacle detection is turned off and bit set indicates that obstacle detection is turned on.

P102 Obstacle position Status
Packet definition: packet length is 21 bits.

b20-b19 = object ID, range is 0 - 3, can represent the positions of up to four objects.

b18-b10 = distance to object from vehicle reference point. Range is 0-511 in decimeters, max range is 51.1 meters.

b9-b0 = angle to object from vehicle reference point. Range is 0-1023 representing 0-359.65 degrees. Forward on the vehicle centerline is 0 degrees with values increasing around the circle in the clockwise direction.

Distribution:

DOE Headquarters
Director, Office of Security Evaluation
U. S. Department of Energy
DP-4
Washington, DC 20545

O. B. Johnson, Deputy Director
Office of Safeguards & Security
U. S. Department of Energy
DP-34.1
Washington, DC 20545

A. Barry Dalinsky, Deputy Director
Policy, Standards & Analysis Division
Office of Safeguards & Security
U. S. Department of Energy
DP-345
Washington, DC 20545

W. J. Desmond, Chief
Physical Security Branch
Office of Safeguards & Security
U. S. Department of Energy
DP-345.1
Washington, DC 20545

W. L. Barker (Acting)
Dep. Asst. Secy. for Security Affairs
DP-30
Washington, DC 20585

E. J. McCallum, Director
Office of Safeguards & Security
U. S. Department of Energy
DP-34
Washington, DC 20545

Willie F. Hensley
Weapons Safeguards & Security Operations
Office of Safeguards & Security
U. S. Department of Energy
DP-346.2
Washington, DC 20545

Nicholas R. Marler
Production & Energy Safeguards
& Security Operations Branch
Office of Safeguards & Security
U. S. Department of Energy
DP-346.3
Washington, DC 20545

Office of Safeguards and Security
Planning & Technology Development Branch
Attn: Judy Young
U. S. Department of Energy
DP-346.4
Washington, DC 20545

Larry D. Wilcher
Technical & Operations Security Branch
Office of Safeguards & Security
U. S. Department of Energy
DP-345.3
Washington, DC 20545

Donald C. Tubbs
Assessment & Integration Branch
Office of Safeguards & Security
U. S. Department of Energy
DP-346.1
Washington, DC 20545

Marshall O. Combs
HQ Physical Protection Branch
Office of Safeguards & Security
U. S. Department of Energy
DP-347.1
Washington, DC 20545

Leon A. Breault, Chief
HQ Technical & Information Security Branch
Office of Safeguards & Security
U. S. Department of Energy
DP-347.3
Washington, DC 20545

K. Sanders, Chief
International Safeguards Branch Office of
Safeguards & Security
U. S. Department of Energy
DP-347.4
Washington, DC 20545

<u>DOE Field Offices</u>

J. E. Anderson, Director
Safeguards & Security Division
U. S. Department of Energy
Savannah River Operations Office
P. O. Box A
Aiken, SC 29802

Acting Chief
Planning & Technology Development
Branch Office of Safeguards & Security
U. S. Department of Energy
DP-346.4
Washington, DC 20545

David Jones, Director HQ Operations
Division Office of Safeguards & Security
U. S. Department of Energy
DP-347
Washington, DC 20545

J. W. Maloney, Manager
Safeguards & Security
Westinghouse Savannah River Company
Savannah River Plant
Aiken, SC 29808-0001

J. L. Loftis, Chief
Operation Branch, Security Division
U. S. Department of Energy
Albuquerque Operations Office
P. O. Box 5400
Albuquerque, NM 87115

Thomas Gradle, Chief
Security Operations Branch
U. S. Department of Energy
Chicago Operations Office
9800 South Cass Avenue
Argonne, IL 60439

Brent Clark, Director
Safeguards & Security Division
U. S. Department of Energy
Idaho Operations Office
785 DOE Place
Idaho Falls, ID 83402

Nicholas Roberts, Chief
Safeguards & Security Division
U. S. Department of Energy
Rocky Flats Area Office
P. O. Box 928
Golden, CO 80401

Michael Hooper, Acting Director
Safeguards and Security Division
U. S. Department of Energy
San Francisco Operations Office
1333 Broadway
Oakland, CA 94612

E. W. Adams, Director
Safeguards & Security Division
U. S. Department of Energy
Nevada Operations Office
P. O. Box 98518
Las Vegas, NV 89193-8518

Robert L. Windus, Security Manager
U. S. Department of Energy
Bonneville Power Administration
P. O. Box 3621
Portland, OR 97208

A. C. Walker, Chief
Safeguards Branch
U. S. Department of Energy
Richland Operations Office
P. O. Box 550
Richland, WA 99352

Branch Chief
U. S. Department of Energy
Physical Security Safeguards & Security Division
Oak Ridge Operations Office
Oak Ridge, TN 37830

Other Govt. Agencies

Doug Carlisle, Manager
Safeguards & Security
Pacific Northwest Lab
P. O. Box 999
Richland, WA 99352

Assistant to the Secretary of Defense
(Atomic Energy)
Room 3E1074, The Pentagon
Washington, DC 20301-3050
  Attn: MIL APPL (Col Johnson) (1)

Defense Advanced RSCH Proj Agency
1400 Wilson Blvd.
Arlington, VA 22209-2308
  Attn: W. Isler - EAO (1)
        STO N. Doherty (1)

Director
Defense Nuclear Agency
Washington, DC 20305-1000
  Attn: TITL (4)
        NSNS (2)
        OPNA (2)
        OPNS (1)
        OPNO (1)

Director
Defense Scty Institute
Richmond, VA 22329-5091
  Attn: Facilities Protection (1)

Defense Technical Information Center
Cameron Station
Alexandria, VA 22304-6145
  Attn: DD (2)

Headquarters
Department of the Army
Washington, DC 20310-0440
  Attn: DAMA-CSC-ST (1)
        DAPE-HRE (1)
        DAMO-NCS (1)
        DAMO-SWS (1)

Deputy Chief of Staff for Personnel
Office of the Office of Army Law Enforcement
Washington, DC 20310-0300
  Attn: DAPE-HRE (1)

Commander
Seneca Army Depot
Romulus, NY 14541-5001
  Attn: Surety Ofc, SDSSE-AW (1)
        Provost Marshal (1)

Commander
Sierra Army Depot
Herlong, CA 96113
  Attn: Security Operations (1)

Commander
U.S. Army Belvoir R&D Ctr
Fort Belvoir, VA 22060-5166
  Attn: STRBE-XI
        AMC-PM-PSE (1)
        STRBE-X (1)
        STRBE-ZK (1)
        STRBE-ZPS (1)
        STRBE-ES (1)
        STRBE-N (1)

U.S. Army Human Engineering Lab
Aberdeen Proving Ground, MD 21005
  Attn: SLCHE-CS (1)
        SLCHE-CC (1)
        Dr. D. Hodge (1)

Commander
U.S. Army Military Police School
Fort McClellan, AL 36205
  Attn: ATZN-MP-TS (1)
        ATZN-MP-CD (1)
        ATZN-MP-DSF-L (1)
        ATZN-MP-DE (1)
        ATZN-MP-TB (1)

Commander
U. S. Army Tank Automotive R&D Cmd.
Warren, MI 48397-5000
  Attn: AMSTA-RCAF (1)

Commander
Naval Ocean Systems Center
San Diego, CA 92152-5000
  Attn: CODE 442 (1)
        CODE 531 (1)

Headquarters
Air Force Office of Security Police
Kirtland AFB, NM 87117
  Attn: AFOSP/SPE (1)
        AFOSP/SPPC (1)
        AFOSP/SPOS (1)

DET 2, HQ SAC/SP
Norton AFB, CA 92409-6468
  Attn: SP (1)

Headquarters
Electronics Systems Div/TCB
Hanscom AFB, MA 01731-5000
  Attn: Physical Sec Sys Dir (1)

Headquarters
Strategic Air Command/SPP
Dept of the Air Force
Offutt AFB, NE 68113
  Attn: SPP (1)

Headquarters
Strategic Air Command/SPO
Dept of the Air Force
Offutt AFB, NE 68113
  Attn: SPO (1)

Dept. of Energy
GTN
Washington, DC 20545
  Attn: DASMA, DP-20 (1)

U.S. Army Armament Research, Development
  and Engineering Center
Picatinny Arsenal, NJ 07806-5000
  Attn: SMCAR-FSN-M
        R. Herman (1)

LtCol Harper
USMC
UGV/JPO
U.S. Army Missile Command (MICOM)
Attn: AMSMI-RD-UG (AMC-PM-UG)
Redstone Arsenal
Huntsville, AL 35898

Sandia National Laboratories Distribution:

1	5200	J. Jacobs
1	5260	J. R. Kelsey
1	5267	S. C. Roehrig
1	5267	J. B. Pletta
15	5267	Div. Files
1	8523	R. C. Christman
5	3141	S. A. Landenberger
8	3145	Document Processing For DOE/OSTI
3	3151	G. C. Claycomb